

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-200255

(43)Date of publication of application : 18.07.2000

(51)Int.Cl.

G06F 15/177

G06F 1/12

G06F 11/18

(21)Application number : 11-001850

(71)Applicant : HITACHI LTD

(22)Date of filing : 07.01.1999

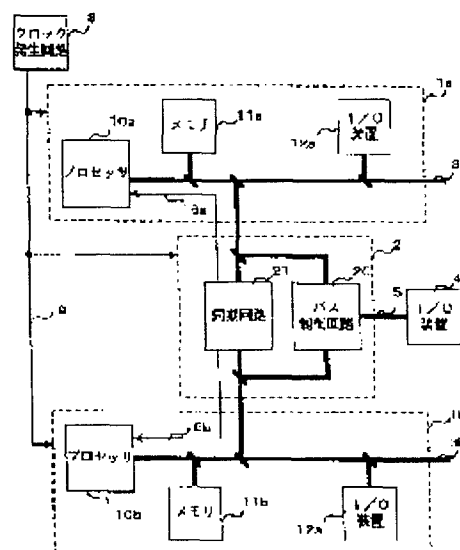
(72)Inventor : TAKEHARA TAKESHI
YAMAGUCHI SHINICHIRO
MIYAZAKI NAOTO
FUJIWARA MICHIO
IKEDA HISAHIO

(54) METHOD AND CIRCUIT FOR SYNCHRONIZATION BETWEEN PROCESSORS

(57)Abstract:

PROBLEM TO BE SOLVED: To easily and speedily synchronize processors in operation in a multiprocessor system.

SOLUTION: Processors 10a and 10b which request themselves to be synchronized with each other a read access to the synchronizing circuit 21. The synchronizing circuit 21 pass answer data showing that the synchronization is successful to the processors when the read access from the processors is completed within a predetermined time. When not, on the other hand, the circuit passes answer data showing that the synchronization ends in failure to the processors.



LEGAL STATUS

[Date of request for examination] 28.05.2001

[Date of sending the examiner's decision of rejection] 10.06.2003

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(51)Int.Cl. ⁷	識別記号	F I	テマコード*(参考)
G 0 6 F 15/177	6 8 0	G 0 6 F 15/177	6 8 0 C 5 B 0 3 4
1/12		11/18	3 1 0 F 5 B 0 4 5
11/18	3 1 0	1/04	3 4 0 A

審査請求 未請求 請求項の数8 O L （全 9 頁）

(21)出願番号	特願平11－1850	(71)出願人	000005108 株式会社日立製作所 東京都千代田区神田駿河台四丁目 6 番地
(22)出願日	平成11年 1 月 7 日(1999.1.7)	(72)発明者	竹原 剛 茨城県日立市大みか町七丁目 1 番 1 号 株 式会社日立製作所日立研究所内
		(72)発明者	山口 伸一郎 茨城県日立市大みか町七丁目 1 番 1 号 株 式会社日立製作所日立研究所内
		(74)代理人	100087170 弁理士 富田 和子

最終頁に続く

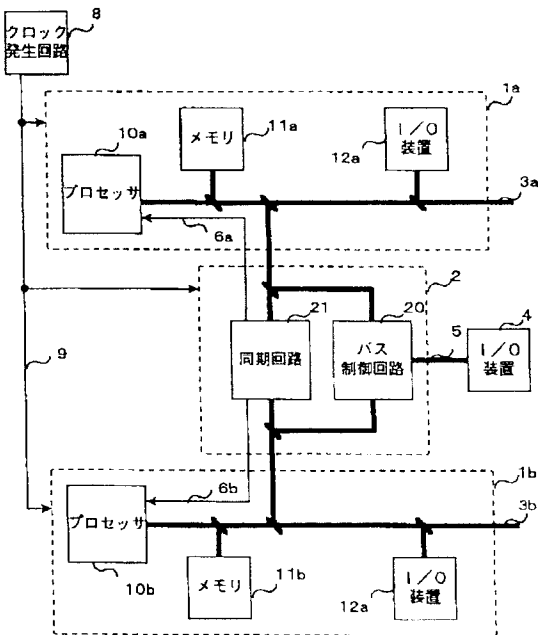
(54)【発明の名称】 プロセッサ間の同期化方法及び同期回路

(57)【要約】

【課題】 多重化システムにおいて、動作中の複数のプロセッサ間の同期化を、容易にかつ迅速に実現する。

【解決手段】 同期化を要求する各プロセッサ 1 0 a、1 0 bは、同期回路 2 1 に対してリードアクセスを行う。同期回路 2 1 は、各プロセッサからのリードアクセスが、予め定められた時間内にあったときは、同期化が成功したことを通知する応答データを各プロセッサに同時に渡す。一方、各プロセッサからのリードアクセスが、予め定められた時間内になかったときは、同期化が失敗したことを通知する応答データを各プロセッサに渡す。

図1



【特許請求の範囲】

【請求項 1】 同一周波数のクロックで動作する複数のプロセッサ間の同期化方法であって、

各プロセッサは、予め定められたアドレスに対し、アクセスを行い、

最初にアクセスを行ったプロセッサのアクセス開始から予め定められた時間内に、他のプロセッサから前記アクセスが行われた場合、最後にアクセスを行ったプロセッサのアクセスに呼応して、各プロセッサに第 1 の応答データを返すことを特徴とする同期化方法。

【請求項 2】 同一周波数のクロックで動作する複数のプロセッサ間の同期化方法であって、

第 1 のプロセッサは、各プロセッサに対し、割込要求を発生させ、

割込要求を受けた各プロセッサは、割込処理として、予め定められたアドレスに対し、アクセスを行い、

最初にアクセスを行ったプロセッサのアクセス開始から予め定められた時間内に、他のプロセッサから前記アクセスが行われた場合、最後にアクセスを行ったプロセッサのアクセスに呼応して、各プロセッサに第 1 の応答データを返すことを特徴とする同期化方法。

【請求項 3】 少なくとも 1 つのプロセッサからのアクセスが、前記予め定められた時間内にない場合、当該時間の経過に呼応して、アクセスを行っているプロセッサに第 2 の応答データを返すことを特徴とする請求項 1 又は請求項 2 に記載の同期化方法。

【請求項 4】 同一周波数のクロックで動作する複数のプロセッサ間を同期させる同期回路であって、

各プロセッサからのアクセスを検知する検知手段と、

各プロセッサからのアクセスが、最初にアクセスを行ったプロセッサのアクセス開始から予め定められた時間内に行われた場合、最後にアクセスを行ったプロセッサのアクセスに呼応して、第 1 の応答データを各プロセッサに返す応答手段とを備えたことを特徴とする同期回路。

【請求項 5】 各プロセッサに対し、割込要求を発生させる割込生成手段を更に備えたことを特徴とする請求項 4 に記載の同期回路。

【請求項 6】 前記応答手段は、

少なくとも 1 つのプロセッサからのアクセスが、前記予め定められた時間内にない場合、当該時間の経過に呼応して、アクセスを行っているプロセッサに第 2 の応答データを返すことを特徴とする請求項 4 又は請求項 5 に記載の同期回路。

【請求項 7】 前記応答手段は、

前記応答データを返すまでの間、アクセスを行ったプロセッサに対してデータ待ち信号を出力することを特徴とする請求項 4 ～ 6 のいずれか一項に記載の同期回路。

【請求項 8】 同一周波数のクロックで動作する複数のプロセッサを備えた多重化システムであって、

請求項 4 ～ 7 のいずれか一項に記載の同期回路を備えた

ことを特徴とする多重化システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、複数のプロセッサを有する多重化システムに関し、特に、同一クロックで動作する複数のプロセッサに同一の処理を実行させるために、プロセッサ間を同期させる方式に関する。

【0002】

【従来の技術】従来、システムの信頼性を向上させるため、複数のプロセッサを多重化し、同一の処理を同期して実行させるシステムが知られている。このようなシステムにおいては、例えば、故障の発生に対する回復処理後に、再度多重化動作を行わせるために、システム動作中における同期化方法が必要となる。

【0003】このような同期化の方法としては、例えば、特開平 08-278950 号公報に記載されている方法がある。本公報には、複数の演算処理装置で構成された多重化コンピュータシステムにおいて、障害から復旧した演算処理装置を再同期させる際に、正常動作している演算処理装置及び故障から回復した演算処理装置より、予め定められたアドレスにアクセス要求を行い、正常動作している演算処理装置からのアクセス要求時刻と、故障から回復した演算処理装置からのアクセス要求時刻との時間差が、予め定められた時間差以内である場合、両演算処理装置が、両演算処理装置が接続される処理装置からアクセス許可を受け取り、アクセス許可を受け取った両演算処理装置からのアクセスを同期化のための基準信号として、両演算処理装置の動作を再同期する障害回復方法が記載されている。

【0004】

【発明が解決しようとする課題】しかしながら、上述した方法では、同期化のための基準信号が得られる時点と、同期化処理が終了するまでの間に時間差があり、その分、再同期に時間がかかる。

【0005】本発明の目的は、システム動作中の複数のプロセッサ間の同期化を、容易にかつ迅速に実現する同期化方法及び同期回路を提供することにある。

【0006】

【課題を解決するための手段】本発明に係る同期化方法は、同一周波数のクロックで動作する複数のプロセッサ間の同期化方法である。そして、本発明に係る第 1 の同期化方法は、各プロセッサは、予め定められたアドレスに対し、アクセス（例えば、リードアクセス）を行い、最初にアクセスを行ったプロセッサのアクセス開始から予め定められた時間内に、他のプロセッサから前記アクセスが行われた場合、最後にアクセスを行ったプロセッサのアクセスに呼応して、各プロセッサに第 1 の応答データを返すことを特徴とする。

【0007】前記第 1 の応答データは、各プロセッサに対して、例えば、前記複数のプロセッサのすべてのプロ

3

セッサ間で同期化が成功したことを知らせるデータである。

【0008】また、本発明に係る第2の同期化方法は、第1のプロセッサは、各プロセッサに対し、割込要求を発生させ、割込要求を受けた各プロセッサは、割込処理として、予め定められたアドレスに対し、アクセスを行い、最初にアクセスを行ったプロセッサのアクセス開始から予め定められた時間内に、他のプロセッサからアクセスが行われた場合、最後にアクセスを行ったプロセッサのアクセスに呼応して、各プロセッサに第1の応答データ

を返すことを特徴とする。

【0009】この場合において、少なくとも1つのプロセッサからのアクセスが、前記予め定められた時間内がない場合、当該時間の経過に呼応して、アクセスを行っているプロセッサに第2の応答データを返すようにしてもよい。

【0010】この第2の応答データは、各プロセッサに対して、例えば、少なくとも1つのプロセッサについては同期化が成功しなかったことを知らせるデータである。第2の応答データを受け取った各プロセッサは、同期処理を中断して、他の処理を行うようにしてもよい。

【0011】本発明に係る同期回路は、同一周波数のクロックで動作する複数のプロセッサ間を同期させる同期回路である。そして、本発明に係る第1の同期回路は、各プロセッサからのアクセスを検知する検知手段と、各プロセッサからのアクセスが、最初にアクセスを行ったプロセッサのアクセス開始から予め定められた時間内に行われた場合に、最後にアクセスを行ったプロセッサのアクセスに呼応して、第1の応答データを各プロセッサに返す応答手段とを備えたことを特徴とする。

【0012】この場合において、各プロセッサに対し、割込要求を発生させる割込生成手段を更に備えるようにしてもよい。この割込生成手段は、例えば、各プロセッサからのアクセス（例えば、ライトアクセス）を受けると、各プロセッサに対し、割込要求を発生するようにしてもよい。

【0013】また、前記応答手段は、少なくとも1つのプロセッサからのアクセスが、予め定められた時間内がない場合、当該時間の経過に呼応して、第2の応答データをアクセスを行っているプロセッサに返すようにしてもよい。また、前記応答データを返すまでの間、アクセスを行ったプロセッサに対してデータ待ち信号を出力するようにしてもよい。

【0014】本発明に係る多重化システムは、同一周波数のクロックで動作する複数のプロセッサを備えた多重化システムであって、前述した同期回路を備えたことを特徴とする。

【0015】

【発明の実施の形態】以下、本発明の実施の形態について、図面を参照しながら説明する。

4

【0016】図1は、本発明を適用した二重化システムの構成を示す図である。

【0017】同図に示すように、本システムは、A系システム1aと、B系システム1bと、バス制御部2と、I/O装置4と、クロック発生回路8を備える。

【0018】A系システム1a及びB系システム1bは、それぞれ、バス3a及び3bを介して、バス制御部2に接続されている。バス制御部2とI/O装置4とは、共通バス5によって接続されている。

【0019】クロック発生回路8は、A系システム1a、B系システム1b及びバス制御部2に対し、同一のクロック9を供給する。A系システム1a、B系システム1b及びバス制御部2は、クロック9に同期して動作する。

【0020】A系システム1aは、所定の演算処理を実行するプロセッサ10aと、プロセッサの処理プログラムや、演算データを記憶するメモリ11aと、プロセッサの命令により所定の動作を実行するI/O装置12aとから構成され、各構成要素はバス3aに接続されている。

【0021】B系システム1bは、A系システム1aと同一の構成をとっており、プロセッサ10bとメモリ11bとI/O装置12aとから構成され、各構成要素はバス3bに接続されている。

【0022】バス制御部2は、バス制御回路20と、同期回路21とから構成され、いずれの構成要素もバス3a、3bに接続されている。

【0023】バス制御回路20は、両系バス3a、3bの信号の比較照合や、両系システム1a、1bから共通バス5上のI/O装置4に対するアクセスの制御を行っている。

【0024】A系プロセッサ10a及びB系プロセッサ10bは、同一の処理を同期して実行しており、バス制御回路20は、両系バス3a、3bの信号の比較照合することにより、両系システム1a、1bにおいて、同一の処理が同期して実行されているかを監視する。

【0025】同期回路21は、両系プロセッサ10a、10bからリードアクセスを受け、そのアクセスに対する応答データを両系に同時に返すことにより両系プロセッサの同期化を実現する。

【0026】同期回路21は、両系プロセッサ10a、10bからのアクセスができるように、例えば、各プロセッサ10a、10bのアドレス空間の所定の領域にマッピングされている。また、同期回路21は、応答データを両系に同時に返すために、ウェイト信号6a、6bを使って、プロセッサの状態を制御している。なお、ウェイト信号6a、6bは、バス3a、3bにおける制御信号の一種である。

【0027】図1の二重化システムでは、両系プロセッサ10a、10bに、同一の処理を実行させているとき

5

に、例えば、故障等の影響で同期ずれが発生すると、所定のエラー回復処理を行った後、両系プロセッサ 10 a、10 b を再度、同期化させるため、同期回路 21 にアクセスさせる。これにより、プロセッサ間を容易に同期させることができる。

【0028】すなわち、同期回路 21 は、各プロセッサからのアクセス時刻の時間差が、予め定められた値以内である場合に、特定の応答データを各プロセッサに同時に返すことによりプロセッサ間の同期を実現する。

【0029】図 2 は、図 1 に示した同期回路 21 の構成を示す図である。

【0030】同図に示すように、同期回路 21 は、デコーダ 211 a、211 b と、各系からの同期化要求の有無を保持するフラグ 212 a、212 b と、ウェイト制御回路 210 a、210 b と、AND ゲート 213 とから構成される。

【0031】デコーダ 211 a、211 b は、バス 3 a、3 b 上に出力されたアドレス及び制御信号から、同期回路 21 に対するアクセスを検知すると、同期化要求信号 215 a、215 b を出力する。

【0032】フラグ 212 a、212 b は、同期化要求信号 215 a、215 b がアサートされると、論理値 1 をクロックに同期して保持する。AND ゲート 213 は、フラグ 212 a、212 b の値の論理積をとり、同期化信号 216 として出力する。すなわち、同期化信号 216 は、同期回路 21 へのリードアクセスが両系から実行されているときに、論理値 1 となる。従って、同期化信号 216 は、両系からのアクセスに対する応答を同時に実行するための、基準信号となる。

【0033】また、ウェイト制御回路 210 a、210 b は、同期化要求信号 215 a、215 b と、同期化信号 216 とから、ウェイト信号 6 a、6 b を生成するとともに、同期回路 21 へのリードアクセスに対する応答データをバス 3 a、3 b 上に出力する。

【0034】図 3 は、図 2 に示したウェイト制御回路 210 a の内部構成を示す図である。なお、ウェイト制御回路 210 b も、ウェイト制御回路 210 a と同一の構成である。

【0035】同図に示すように、ウェイト制御回路 210 a は、データ出力部 2100 a と、タイマ 2101 a と、AND ゲート 2102 a とから構成される。

【0036】データ出力部 2100 a は、同期化信号 216 がアサートされていると、データ値「1」を、それ以外のときにはデータ値「0」を、バス 3 a 上に応答データとして出力する。

【0037】また、タイマ 2101 a は、同期化要求信号 215 a の立上りにより、時間の計測を開始し、予め定められたタイムアップ時間になると、タイムアップ信号 2103 a をアサートする。

【0038】また、AND ゲート 2102 a は、同期化

6

信号 216 の否定と、同期化要求信号 215 a と、タイムアップ信号 2103 a の否定との論理積をとり、ウェイト信号 6 a として出力する。

【0039】つまり、ウェイト信号 6 a は、（通常、同期化信号 216 とタイムアップ信号 2103 a は、ネゲートされているので、）同期化要求信号 215 a のアサートと同時にアサートされ、同期回路 21 へのアクセスが両系から実行されて同期化信号 216 がアサートされるか、または、予め定められた時間だけウェイト信号 6 a がアサートされ続けてタイムアップ信号 2103 a がアサートされたときに、ネゲートされる。

【0040】次に、同期回路 21 の動作について説明する。

【0041】図 4 は、同期化成功時の同期回路 21 の動作を示すタイムチャートである。

【0042】同図に示すように、まず、A 系システム 1 a のプロセッサ 10 a が同期回路 21 に対してリードアクセスを実行すると、t1 サイクルで、同期化要求信号 215 a がアサートされる。

【0043】これにより、次の t2 サイクルで、フラグ 212 a に論理値 1 がセットされる。また、t1 サイクルでウェイト信号 6 a がアサートされ、次の t2 サイクルでプロセッサ 10 a がウェイト状態になり、同期化要求信号 215 a もアサートされ続ける。t3 サイクルも t2 サイクルと同様である。

【0044】そして、B 系システム 1 b のプロセッサ 10 b が同期回路 21 に対してリードアクセスを行い、t4 サイクルで、同期化要求信号 215 b がアサートされると、次の t5 サイクルでフラグ 212 b に論理値 1 がセットされる。また、t4 サイクルでウェイト信号 6 b がアサートされ、次の t5 サイクルでプロセッサ 10 b がウェイト状態になり、同期化要求信号 215 b がアサートされ続ける。

【0045】また、t5 サイクルで、フラグ 212 a、212 b の値がいずれも論理値 1 となり、同期化信号 216 がアサートされ、ウェイト信号 6 a、6 b がともにネゲートされる。その結果、次の t6 サイクルでは、同期化要求信号 215 a、215 b がともにネゲートされ、ここでプロセッサ 10 a、10 b の同期化が実現される。また、t5 サイクルで、応答データとして、

「1」がバス 3 a、3 b 上に出力され、同期化が成功したことが、プロセッサ 10 a、10 b に通知される。

【0046】図 5 は、同期化失敗時の同期回路 21 の動作を示すタイムチャートである。

【0047】同図に示すように、まず、A 系システム 1 a のプロセッサ 10 a が同期回路 21 に対してリードアクセスを実行すると、t1 サイクルで、同期化要求信号 215 a がアサートされる。

【0048】これにより、次のサイクル t2 からフラグ 212 a に論理値 1 がセットされる。また、t1 サイクル

ルでウェイト信号6 aがアサートされ、次のサイクルからプロセッサ10 aがウェイト状態になり、同期化要求信号215 aもアサートされ続ける。

【0049】そして、予め定められた時間t xだけ時間が経過しても、B系システム1 bのプロセッサ10 bからのアクセスが実行されないと、t 2サイクルでタイムアップ信号2103 aがアサートされ、ウェイト信号6 aがネゲートされる。その結果、次のt 3サイクルで、同期化要求信号215 aがネゲートされる。この場合、同期化信号216 aがアサートされないため、応答データとしてバス3 aに「0」が出力され、同期に失敗したことが、プロセッサ10 aに通知される。

【0050】次に、上述したシステム構成を用いた同期化方法について説明する。

【0051】図6は、プロセッサ10 a、10 b間を同期させるために各プロセッサが実行する同期処理のフローチャートを示す図である。同図に示すように、各プロセッサは、まず、同期回路21にリードアクセスを行う(S 80)。そして、リードアクセスに対する応答データの値を判別する(S 81)。その結果、応答データの値が「0」のときは、同期化に失敗しているので、同期化に成功するまで同期回路21へのアクセスを繰り返す(S 80)。一方、応答データの値が「1」のときは、同期化が成功しているので、同期処理を終了する。プロセッサ10 a、10 bは、それぞれ、このような簡単な処理を実行することにより、プロセッサ間の同期を実現することができる。

【0052】図7は、図1に示した二重化システムにおいて、エラー発生後に回復処理を行って二重化処理を再開するまでの各プロセッサの処理を示すフローチャートである。

【0053】システムが正常に二重化処理を行っている間は、バス制御回路2によるバス比較の結果は一致している。しかし、片側のプロセッサの処理が何らかの原因で一時的に変化して、バス比較の結果が不一致となると、システムはエラー発生を検出する。エラー発生検出後、そのエラーが回復可能であれば、システムでは所定の回復処理を行い、通常処理を再開する。そのために、各プロセッサでは、回避していたデータの復帰などの処理を実行した後、再同期あわせを行うため、図6に示した同期処理を行う。

【0054】すなわち、まず、各プロセッサは、同期回路21に対してリードアクセスを行い、リードアクセスに対する同期回路21の応答データが「0」の時は、同期回路20へのリードを繰り返し実行する。そして、リードアクセスに対する同期回路20の応答データが「1」になった時、同期化が成功したことを検知し、二重化処理を再開する。以上のように、本実施形態においては、同期ずれが発生した時の再同期化が容易に行える。

【0055】図8は、プロセッサ間を同期させるときの各プロセッサにおける処理の様子を示す図である。

【0056】同図に示すように、同期化されていない状態で、処理A1を実行したプロセッサ10 aは、次の処理で、プロセッサ間を同期させるために、第一回目の同期処理を開始する。このとき、プロセッサ10 bは、処理B1、処理B2を実行しているため、プロセッサ10 aの第一回目の同期処理による同期化は成功しない。

【0057】プロセッサ10 aは、続いて、第二回目の同期処理を開始する。プロセッサ10 aが第二回目の同期処理を実行している最中に、プロセッサ10 bも、同期処理を実行し、ここに同期化が成功する。これにより、以降の処理では、プロセッサ10 a、10 bは、同一の処理を同期実行することが可能になる。

【0058】なお、以上の説明においては、ウェイト信号6 a、6 bを使って、応答データを両系に同時に返す制御を行っているが、これに限らず、バスの構成にあわせて、他の信号(例えば、データアクリッジ信号)によって応答データを両系に同時に返す制御を行ってもよい。

【0059】これまで説明した多重化システムでは、正常時は、A系プロセッサ10 a及びB系プロセッサ10 bは、同一の処理を実行していた。しかし、多重化システムにおいて、より複雑な動作を実現するため、各プロセッサに異なる動作を行わせて、信頼性や安全性を要する処理を実行するときだけ同期化して同一の処理を行わせることも考えられる。

【0060】このように、全く異なる動作をしているプロセッサ間を同期させる場合でも、前述した方法を適用することはできるが、前述したシステム構成では、同期化に長い時間がかかる場合も生じうる。そこで、次に、このような場合により適した二重化システムについて説明する。

【0061】図9は、本発明を適用した第二の二重化システムの構成を示す図である。本システムでは、各プロセッサが全く非同期に動作しているときでも、同期化を短時間で成功させることが可能になる。

【0062】図9に示したシステムは、バス制御部2に、割込生成回路23が追加されている以外は、図1に示したシステムと同一の構成である。

【0063】割込生成回路23は、バス3 a、3 bに接続されており、プロセッサ10 a、10 bからアクセス(例えば、ライトアクセス)を受けると、プロセッサ10 a、10 bに対して、それぞれ、割込要求信号7 a、7 bを出力する。

【0064】割込要求信号7 a、7 bを受けたプロセッサ10 a、10 bは、所定の割込処理を実行する。

【0065】図10は、割込生成回路23の構成を示す図である。

【0066】同図に示すように、割込生成回路23は、

デコーダ 230a、230b と、割込パルス生成回路 231a、231b と、OR ゲート 232a、232b とから構成される。

【0067】デコーダ 230a、230b は、バス 3a、3b 上に出力されたアドレス及び制御信号から、割込生成回路 23 に対するアクセスを検知すると、アクセス要求信号 233a、233b を出力する。

【0068】割込パルス生成回路 231a、231b は、アクセス要求信号 233a、233b がアサートされると、割込パルス 234a、234b を出力する。割込パルス 234a、234b は、OR ゲート 232a、232b を介して、割込要求信号 7a、7b を、プロセッサ 10a、10b の両方に出力する。

【0069】図 11 (b) は、図 9 に示したシステムにおいて、全く異なる動作をしているプロセッサ間を同期させるときの各プロセッサの処理の様子を示す図である。比較のため、図 11 (a) には、同じ状況下で、図 1 に示したシステムにおいて、プロセッサ間を同期させるときの各プロセッサの処理の様子を示している。

【0070】図 11 (a) において、プロセッサ 10a は、処理 A1 の次の処理で、同期処理を開始する。しかし、プロセッサ 10b では、処理 B1～処理 B4 を実行した後に同期処理を開始するため、プロセッサ 10a は、第三回目の同期処理を実行するまで同期化に失敗する。つまり、同期化が成功するまでに、少なくともタイム 2101a がタイムアップする時間の 2 倍の時間がかかっていることになる。

【0071】一方、図 11 (b) においては、プロセッサ 10a は、処理 A1 の次の処理で、割込生成回路 23 へアクセスを実行して割込生成を行っている。割込処理として、図 6 の同期処理を指定しておく、プロセッサ 10a は、割込要求信号 7a を受けて、同期処理を開始する。また、プロセッサ 10b も、処理 B2 を実行中に割込要求信号 7b がアサートされ、処理 B2 の次の処理が割込処理となり、同期処理が開始される。つまり、この場合、プロセッサ 10a は、第一回目の同期処理で同期化に成功することになる。

【0072】このように、図 9 の割込生成回路 23 を利用すれば、各プロセッサが全く異なる動作をしている場合であっても、迅速に同期化が実現できる。

【0073】なお、以上の説明においては、二重化システムを例として説明していたが、より多くのプロセッサを多重化したシステムに対して本発明を適用することも可能である。この場合、複数のプロセッサから同期回路に対しアクセスを実行させ、全てのプロセッサからアクセスが実行されているときに、同期化が成功したことを通知する応答データを同時に渡すようにすればよい。

【0074】また、以上の説明においては、各プロセッサには、同一位相のクロックを入力していたが、位相の異なる（例えば、クロック位相が半サイクルずれた）同

一周波数のクロックを入力するようにしてもよい。この場合、バス制御回路 20 は、位相のずれを考慮して、バス 3a、3b の信号の位相を同期させてから、信号の比較照合等を行う。また、同期回路 21 は、位相のずれを考慮した各プロセッサ毎のタイミングで、各プロセッサに応答データを返す。

【0075】

【発明の効果】以上詳細に説明したように、本発明によれば、同期回路が複数のプロセッサからのリード・アクセスに対し、特定の応答データを特定のタイミング（例えば、同時に）渡すことにより、容易にプロセッサ間を同期させることができる。さらに、割込生成回路を用いて、同期回路へのアクセスを割込処理として実行することにより、迅速にプロセッサ間を同期させることができる。

【図面の簡単な説明】

【図 1】本発明による二重化システムの構成を示す図である。

【図 2】同期回路 21 の構成を示す図である。

【図 3】ウェイト制御回路 210a の構成を示す図である。

【図 4】同期化成功時の同期回路 21 の動作を示すタイムチャートである。

【図 5】同期化失敗時の同期回路 21 の動作を示すタイムチャートである。

【図 6】同期化のために、各プロセッサが実行する同期処理のフローチャートを示す図である。

【図 7】二重化システムにおいて、エラー発生後に二重化処理を再開するまでのフローチャートである。

【図 8】プロセッサ間の同期化を行うときのプロセッサ 10a、10b の処理の様子を示した図である。

【図 9】本発明による第二の二重化システムの構成を示す図である。

【図 10】割込生成回路 23 の構成を示す図である。

【図 11】プロセッサ間の同期化を行うときのプロセッサ 10a、10b の処理の様子を示した図である。

【符号の説明】

1a…A系システム

1b…B系システム

2…バス制御部

3a、3b…バス

4…I/O装置

5…共通バス

6a、6b…ウェイト信号

8…クロック発生回路

9…クロック信号

10a、10b…プロセッサ

11a、11b…メモリ

12a、12b…I/O装置

20…バス制御回路

10

20

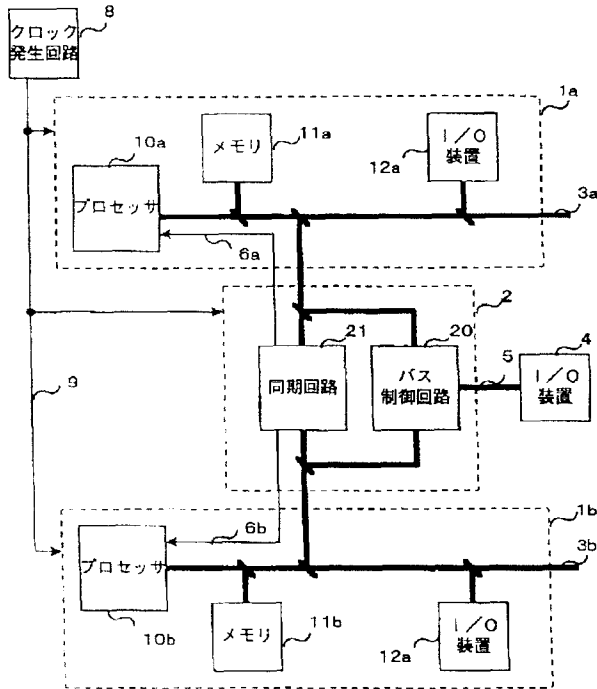
30

40

50

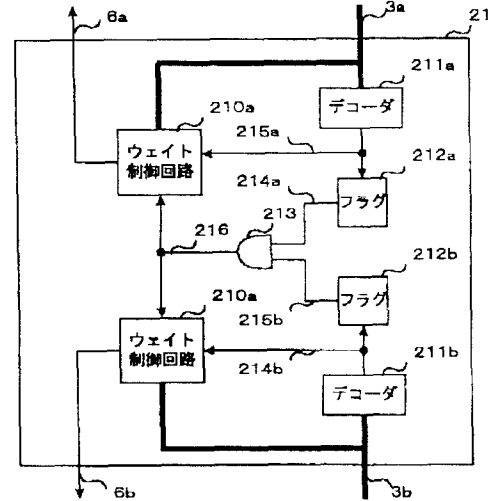
【図1】

図1



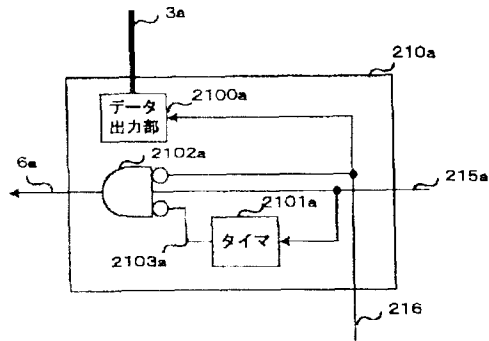
【図2】

図2



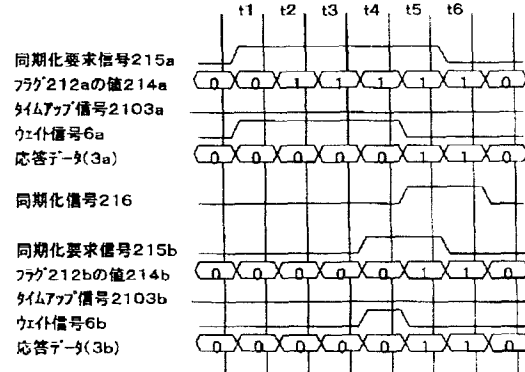
【図3】

図3



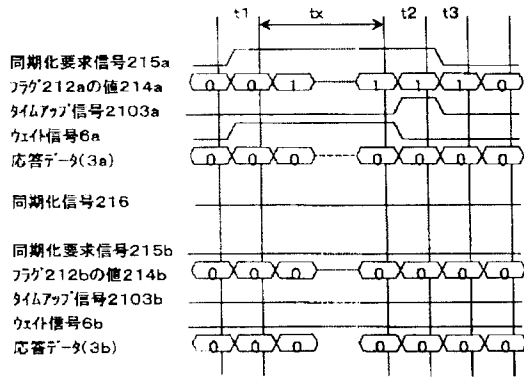
【図4】

図4



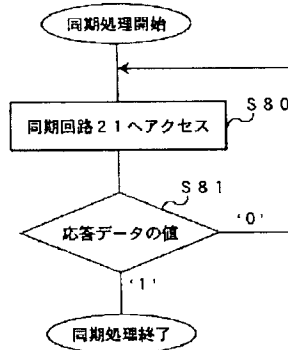
【図5】

図5



【図6】

図6

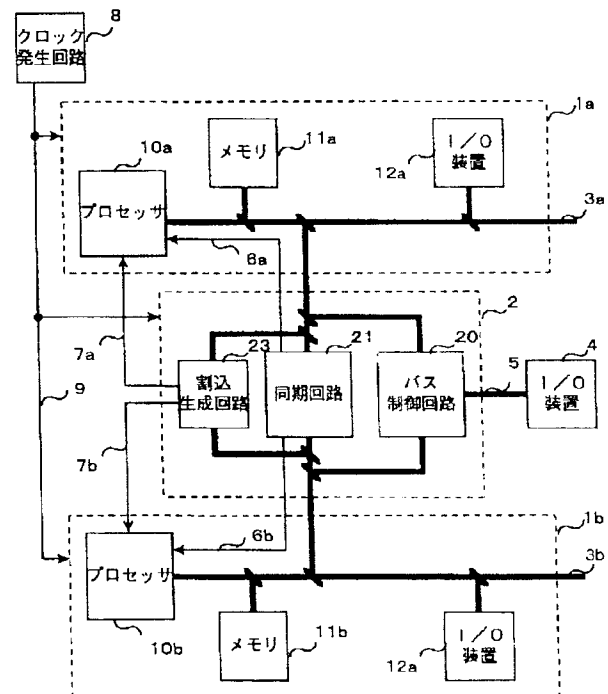
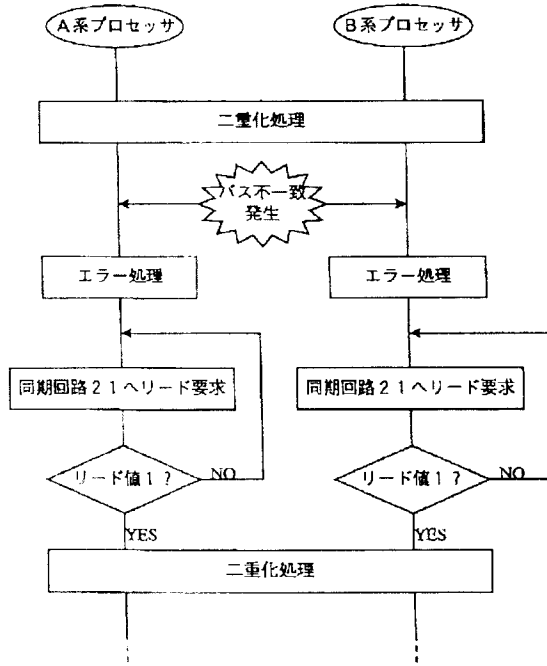


【図9】

図9

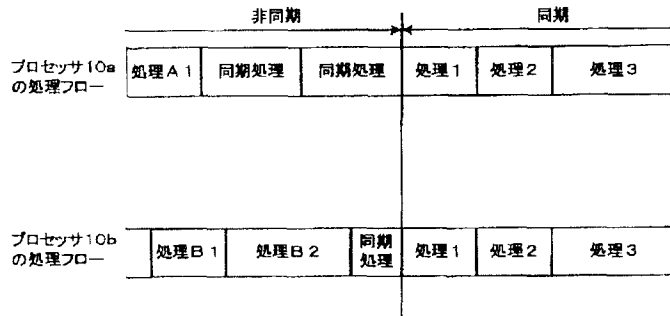
【図7】

図7



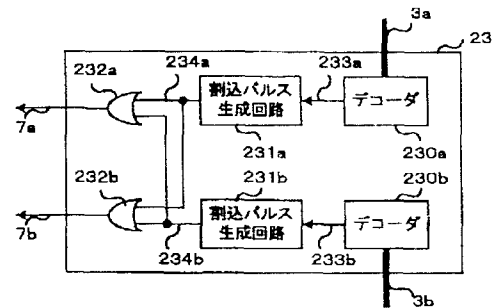
【図8】

図8



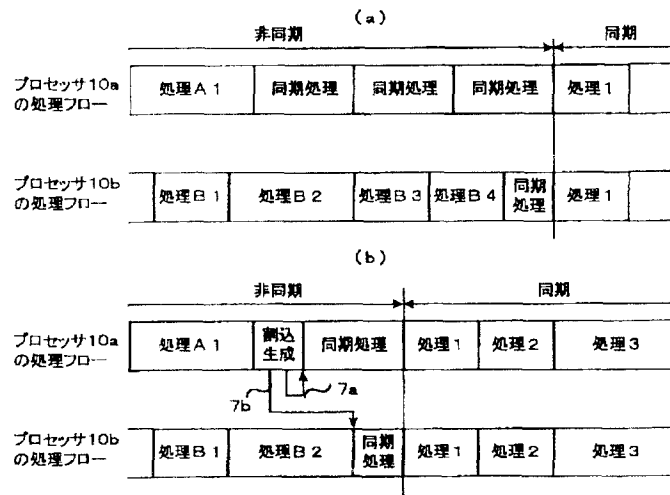
【図10】

図10



【図11】

図11



フロントページの続き

(72)発明者 宮崎 直人
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内
(72)発明者 藤原 道雄
茨城県ひたちなか市市毛1070番地 株式会
社日立製作所水戸工場内

(72)発明者 池田 尚弘
茨城県ひたちなか市市毛1070番地 株式会
社日立製作所水戸工場内
Fターム(参考) 5B034 AA04 CC01 DD06
5B045 CC02 CC08 FF03